An Improved Algorithm for Robust Safety Analysis of Sampled Data Systems

Ian M. Mitchell[1]
[1]Department of Computer Science
University of British Columbia
mitchell@cs.ubc.ca

Shahab Kaynama[1,2]
[2]Department of Electrical Engineering &
Computer Science
University of California, Berkeley
kaynama@ece.ubc.ca

ABSTRACT

A sampled data model falls somewhere between continuous and discrete time models: The plant evolves in continuous time, but the controller receives feedback and can modify its control input(s) only at periodic points in time. In previous work we have demonstrated how to compute the discriminating kernel (also called the maximal robust control invariant set) for sampled data systems and how this kernel can be used to analyze and even synthesize safe feedback controllers for systems with state space safety constraints; however, the algorithm for computing the kernel was conservative. In this paper we provide an improved abstract algorithm whose computations are tight to the sampled data discriminating kernel. The improved algorithm can also take sample time jitter into account. A level set implementation is used to demonstrate that the new algorithm is tight and a conservative ellipsoidal implementation is used to demonstrate its practical benefits on a nonlinear quadrotor model.

Categories and Subject Descriptors

I.6.4 [**Simulation and Modeling**]: Model Validation and Analysis; G.1.10 [**Numerical Analysis**]: Applications

Keywords

sampled data, jitter, robust safety analysis, reachability, viability, Hamilton-Jacobi equations, ellipsoids

1. INTRODUCTION

Sampled data is a useful modelling paradigm for cyber-physical systems because it captures key properties of a common design pattern: a continuous time and continuous state plant is attached through sensors and actuators to a discrete time controller. The controller receives state feedback at discrete sample times and generates a control input which is applied to the plant through a zero order hold actuator. Unlike discrete time models, a sampled data model takes into account the trajectory of the plant between sample times.

Maintaining safety is a common challenge in the design of controllers for cyber-physical systems, and it typically appears in the form of constraints on the system state and control input value. The discriminating kernel (or robust control invariant set) is the set of states from which at least one control input signal gives rise to a trajectory which remains inside the state constraint despite the actions of disturbance inputs. In [17] we described an abstract algorithm to conservatively approximate the finite horizon discriminating kernel for sampled data systems and then demonstrated how the results can be used to construct a set-valued control policy which ensures finite horizon safety. We also provided two concrete implementations of the abstract algorithm: one based on Hamilton-Jacobi (HJ) partial differential equations (PDEs) which handles nonlinear dynamics but which is not dimensionally scalable, and a second based on ellipsoidal reachability which requires linear time invariant (LTI) dynamics but which scales well with dimension.

In this paper we propose an improved algorithm for computing sampled data discriminating kernels. The contributions are:

- A proof that the improved algorithm is tight to the sampled data discriminating kernel for systems with fixed sample time sequences.
- A proof that the improved algorithm can compute discriminating kernels which are robust to uncertainty in the sample time sequence; in other words, it can conservatively approximate the kernel for systems that are subject to timing jitter.
- An empirically derived nonlinear model of quadrotor height maintenance is used to show that the ellipsoidal implementation of the improved algorithm is faster and more accurate than the original algorithm, and that the ellipsoidal implementation can generate usable control sets for a realistic model despite the inherent accuracy limitations of ellipsoidal set representations and LTI dynamics.

1.1 Problem Definition

We assume a Markovian system with state $x \in \Omega$, where the state space $\Omega \subseteq \mathbb{R}^{d_x}$ (or some similar vector space of dimension d_x). The system's evolution is modeled by an ordinary differential equation (ODE) with initial condition $x(0) = x_0$ and dynamics

$$\dot{x} = f(x, u, v), \tag{1}$$

where $f : \Omega \times \mathbb{U} \times \mathcal{V} \to \Omega$ is Lipschitz continuous in x and continuous in u and v, $u \in \mathcal{U}$ is the control input, $v \in \mathcal{V}$ is the disturbance input and $\mathcal{U} \subset \mathbb{U} \subseteq \mathbb{R}^{d_u}$ and $\mathcal{V} \subset \mathbb{R}^{d_v}$ are

HSCC '15 April 14–16 2015, Seattle, WA, USA

ACM 978-1-4503-3433-4/15/04?$15.00.
http://dx.doi.org/10.1145/2728606.2728619.

assumed to be compact. Input u seeks to keep the system within the state constraints, while input v seeks to drive the system outside the constraints. The disturbance can be used to model uncertainty or error in f in a worst case fashion so as provide a robust safety analysis.

The sampled data system evolves in continuous time and state according to (1), but the controller only receives state feedback and can set the control signal at sampling times t_k in the sequence $\mathcal{T} = \{t_k\}_{k=0}^{N}$. In order to handle jitter, we divide the time between samples into two components

$$t_{k+1} - t_k = \delta^{\mathrm{F}} + \delta_k^{\mathrm{J}} \tag{2}$$

where $\delta^{\mathrm{F}} \geq 0$ is the fixed minimum time between samples, $\delta_k^{\mathrm{J}} \in [0, \delta^{\mathrm{J}}]$ is the jitter for sample period k, and $\delta^{\mathrm{J}} \geq 0$ is a bound on the jitter. The actual sequence of sample times $\mathcal{T}$ is not known a priori, so in order to characterize the viability constructs discussed in this paper we define the set of feasible sample time sequences

$$\mathbb{T} = \left\{ \mathcal{T} = \{t_k\}_{k=0}^{N} \,\middle|\, \forall t_k \in \mathcal{T}, t_k \in \left[t_{k-1} + \delta^{\mathrm{F}}, t_{k-1} + \delta^{\mathrm{F}} + \delta^{\mathrm{J}} \right] \right\}. \tag{3}$$

For a given sequence of sample times, the dynamics of the system take the form

$$\dot{x}(t) = f(x(t), u_{\mathrm{pw}}(t), v(t)) \tag{4}$$

where the piecewise constant input signal $u_{\mathrm{pw}}(\cdot)$ is chosen according to

$$u_{\mathrm{pw}}(t) = u_{\mathrm{fb}}(x(t_k)) \text{ for } t_k \leq t < t_{k+1} \tag{5}$$

and $u_{\mathrm{fb}} : \Omega \to \mathcal{U}$ is a feedback control policy. Note that the feedback control policy cannot take into account the actual sequence of sample times that are encountered; however, because the input is chosen at the sample times and then held constant, the dynamics (4) *cannot* be written in stationary form $\dot{x} = f(x, v)$.

The state constraint $\mathcal{K}_0 \subset \Omega$ that we seek to maintain for safety is assumed to be the complement of an open set [3]. We divide the state space Ω as in [17]. The outermost set is the safety constraint $\mathcal{K}_0$. It contains a series of nested finite horizon safe sets $\mathcal{K}_k$ which identify states which give rise to trajectories which satisfy the safety constraint for at least k sample times; in other words, a minimum horizon of $k\delta^{\mathrm{F}}$, but the actual safe time will depend on the encountered sample time sequence $\mathcal{T}$. The safety guarantee provided by $\mathcal{K}_k$ requires that the control input $u_{\mathrm{fb}}(\cdot)$ be chosen from a subset of $\mathcal{U}$. In this paper we focus on an improved algorithm for computing the finite horizon constrained control safe sets $\mathcal{K}_k$. A discussion of how to compute a subset of states $\mathcal{K}_{\mathrm{free}}$ with no control constraint and the possibility of finding an infinite horizon safe set $\mathcal{K}_\infty$ are available in [17].

1.2 Related Work

Sampled data systems (with and without sample time uncertainty) are a well-studied area of control engineering, but the focus of much of this research has been on traditional control objectives such as stability, observability and controllability; for example, see [6, 19, 20] and the citations within. In [7] the authors do define a finite time, constant control input reachability construct (r-robust reachability) which is then used to build a piecewise constant control strategy that guarantees robust global asymptotic stability despite sample time uncertainty; however, demonstrating this reachability property for a given system requires a Lyapunov-like function and no general algorithm is discussed for finding such a function.

A full discussion of related work on reachability, viability and verification in the context of sampled data systems can be found in [17], so here we mention only those papers most relevant and/or recent. In [22] the authors study sampled data systems subject to a more restrictive class of sample time jitter: the sample times are always within a fixed interval of a periodic sequence of times. They consider nondeterministic hybrid system dynamics but the controller's input can only influence discrete mode switching rules and they assume that system trajectories are known explicitly in the analysis. They then derive necessary and sufficient conditions for a supremal control invariant predicate which is conceptually similar to the discriminating kernel studied here. In [21] the authors present a tool for verifying sampled data systems which combines Taylor models to explore reachability of the continuous states of the plant with an SMT solver to handle the discrete states of a software controller, but the only non-determinism in the model lies in the initial states. In [5] a different algorithm for approximating the sampled data viability kernel for LTI systems is described which uses a polytopic representation of sets. Based on previous experience with both ellipsoidal and polytopic representations, we suspect that the polytopic scheme is more accurate and scalable, but it is unclear how to create a robust version of that scheme to handle discriminating kernels and/or sample time uncertainty. In [2] the authors use a theorem prover to verify (or even synthesize) invariants and control envelopes robust to very general types of parameter variation, including sample time uncertainty. The class of sample time sequences considered here (3) corresponds to the class "upper and lower bounds on sampling time" in that paper, the discriminating kernel algorithm discussed here is an alternative to the use of a theorem prover in constructing their robust "safe invariant," and the set-valued control policy $\mathcal{U}_{\mathrm{ctrl}}$ described in [17] (and repeated here) corresponds to their "control envelope." Finally, the algorithm in [4] approximates a (robust) reach set which is the complement of the (jitter free) discriminating kernel discussed here. This set is then used to ensure collision avoidance in a pursuer-evader game using real robots (including implementation of elements of the algorithm on an embedded microcontroller).

2. IMPROVED ALGORITHM

In this section we provide an improved algorithm for the sampled data discriminating kernel after defining some notation. Key properties of the algorithm's output are then proved, including its conservativeness under jitter and its tightness in the jitter-free case. Finally the algorithm is demonstrated on two toy examples: one from [17] to demonstrate that the jitter-free case is tight and a new one to demonstrate robustness to jitter.

2.1 Preliminary Definitions

The (jitter robust) sampled data discriminating kernel that we seek is defined by

$$\mathsf{Disc}_{\mathsf{sd}}\left([0,T], \mathcal{S}\right) \triangleq \left\{ x_0 \in \mathcal{S} \,\middle|\, \begin{array}{c} \exists u_{\mathrm{pw}}(\cdot), \forall \mathcal{T} \in \mathbb{T}, \forall v(\cdot), \\ \forall t \in [0,T], x(t) \in \mathcal{S} \end{array} \right\}, \tag{6}$$

where $x(\cdot)$ solves (4) with initial condition $x(0) = x_0$. In addition to the robustness required of the sampled data discriminating kernel discussed in [17], this kernel requires that the feedback policy (5) which generates the piecewise constant input signal $u_{\mathrm{pw}}(\cdot)$ must be robust to variation in the sample time sequence $\mathcal{T}$ as characterized by (3). Given the horizon of interest T, define the maximum relevant sample time sequence length

$$\bar{N} \triangleq \left\lceil \frac{T}{\delta^{\mathrm{F}}} \right\rceil$$

and note that N in (3) must be chosen such that $N \geq \bar{N}$.

We repeat a number of definitions from [17]. The approximation of (6) will be performed in an augmented state space

$$\tilde{x} \triangleq \begin{bmatrix} x \\ u \end{bmatrix} \in \tilde{\Omega} \triangleq \Omega \times \mathbb{R}^{d_u}$$

with dynamics

$$\frac{d}{dt}\tilde{x} = \frac{d}{dt}\begin{bmatrix} x \\ u \end{bmatrix} = \begin{bmatrix} f(x,u,v) \\ 0 \end{bmatrix} \triangleq \tilde{f}(\tilde{x}, v). \tag{7}$$

Movement from $\tilde{\Omega}$ back into Ω or $\mathbb{U}$ is accomplished through projection operators

$$\mathsf{Proj}_x\left(\tilde{\mathcal{X}}\right) \triangleq \left\{ x \in \Omega \;\middle|\; \exists u, \begin{bmatrix} x \\ u \end{bmatrix} \in \tilde{\mathcal{X}} \right\}, \tag{8}$$

$$\mathsf{Proj}_u\left(\tilde{\mathcal{X}}, x\right) \triangleq \left\{ u \in \mathcal{U} \;\middle|\; \begin{bmatrix} x \\ u \end{bmatrix} \in \tilde{\mathcal{X}} \right\}, \tag{9}$$

for $\tilde{\mathcal{X}} \subseteq \tilde{\Omega}$ and $x \in \Omega$.

In [17] we defined a general invariance kernel operator which was then used to construct both $\mathcal{K}_k$ and $\mathcal{K}_{\mathrm{free}}$, but in this paper we are not constructing $\mathcal{K}_{\mathrm{free}}$ so we can simplify the invariance kernel definition

$$\mathsf{Inv}\left([t_s, t_f], \mathcal{S}\right) \triangleq \{\tilde{x}(t_s) \in \mathcal{S} \mid \forall v(\cdot), \forall t \in [t_s, t_f], x(t) \in \mathcal{S}\}, \tag{10}$$

In comparison with the definition in [17], this invariance kernel is always applied in the augmented state space $\tilde{\Omega}$ using the augmented dynamics $\tilde{f}$ and input v treated as a disturbance. The improved algorithm will also make use of a robust reach set construct

$$\mathsf{Reach}\left([t_s, t_f], \mathcal{S}\right) \triangleq \{\tilde{x}(t_s) \in \tilde{\Omega} \mid \forall v(\cdot), \tilde{x}(t_f) \in \mathcal{S}\} \tag{11}$$

Note that this construct is *not* a reach tube—it is the set of states from which all possible trajectories will lead to $\mathcal{S}$ at exactly time t_f; however, those trajectories may be outside of $\mathcal{S}$ for earlier times $t_s \leq t < t_f$. In general reach sets are a weak tool for robust safety analysis; for example, for systems with disturbance inputs in their dynamics, the union of reach sets may be a subset of the reach tube and hence reach sets cannot be used to compute an invariance kernel [16]. In the sampled data case, however, this robust reach set will turn out to be useful.

2.2 Abstract Algorithm

We use an iterative algorithm to construct the jitter robust sampled data discriminating kernel. First, we define some notation for intermediate sets

$$\mathcal{I}_1 \triangleq \mathsf{Inv}\left([0, \delta^{\mathrm{F}} + \delta^{\mathrm{J}}], \mathcal{S} \times \mathcal{U}\right), \tag{12}$$

$$\mathcal{R}_j \triangleq \mathsf{Reach}\left([0, \delta^{\mathrm{F}}], \mathsf{Disc}_{j-1}\left(\mathcal{S}\right) \times \mathcal{U}\right), \tag{13}$$

$$\mathcal{I}_j \triangleq \mathsf{Inv}\left([0, \delta^{\mathrm{J}}], \mathcal{R}_j\right) \tag{14}$$

where $j = 2, 3, \ldots, \bar{N}$ in (13) and (14). Intuitively

- $\mathcal{I}_1$ is the set of states and corresponding constant control values from which (no matter what the disturbance $v(\cdot)$) the trajectory will stay within $\mathcal{S}$ for an entire maximum length sample period.
- $\mathcal{R}_j$ is the set of states and corresponding constant control values from which (no matter what the disturbance $v(\cdot)$) the trajectory will be in $\mathsf{Disc}_{j-1}\left(\mathcal{S}\right)$ in exactly δ^{F} time units.
- $\mathcal{I}_j$ is the set of states and corresponding constant control values from which (no matter what the disturbance $v(\cdot)$) the trajectory will remain in $\mathsf{Proj}_x\left(\mathcal{R}_j\right)$ for the next δ^{J} time units, which from (13) implies that the trajectory will be in $\mathsf{Disc}_{j-1}\left(\mathcal{S}\right)$ during the time interval $[\delta^{\mathrm{F}}, \delta^{\mathrm{F}} + \delta^{\mathrm{J}}]$.

Because there is no projection step between $\mathcal{R}_j$ and $\mathcal{I}_j$, the control value remains fixed over the entire corresponding time interval. We also define

$$\widehat{\mathcal{I}}_j \triangleq \mathcal{I}_j \cap \mathcal{I}_1; \tag{15}$$

for $j = 1, 2, \ldots, \bar{N}$. The algorithm's output is a sequence of sets approximating the sampled data discriminating kernel over various horizons

$$\mathsf{Disc}_j\left(\mathcal{S}\right) \triangleq \mathsf{Proj}_x\left(\widehat{\mathcal{I}}_j\right) \tag{16}$$

for $j = 1, 2, \ldots, \bar{N}$.

At this point we note that the algorithm from [17] was designed to treat the case of time samples with fixed period δ, which corresponds to $\delta^{\mathrm{F}} = \delta$ and $\delta^{\mathrm{J}} = 0$ in the notation of this paper. However, because the old algorithm used invariance kernels for all set evolution, it turns out to be equivalent to the new algorithm with $\delta^{\mathrm{J}} = \delta$ and $\delta^{\mathrm{F}} = 0$; in other words, the sets produced by the old algorithm were in fact robust to any amount of sample jitter up to and including the entire sample period. In light of this observation, it is not surprising that the old algorithm was conservative for the jitter-free case we intended to treat in [17].

In order to solve the problem from section 1.1, we define the finite horizon safe sets in the same way as [17]

$$\mathcal{K}_j = \mathsf{Disc}_j\left(\mathcal{K}_0\right). \tag{17}$$

The control policy which maintains safety is typically set-valued. For $x \in \mathcal{K}_0$, define the safety horizon of x as

$$n(x) \triangleq \begin{cases} \bar{N}, & \text{if } x \in \mathcal{K}_{\bar{N}}; \\ j, & \text{if } x \in \mathcal{K}_j \setminus \mathcal{K}_{j+1}; \end{cases} \tag{18}$$

for $j = \bar{N} - 1, \bar{N} - 2, \ldots, 0$. The control policy is given by

$$\mathcal{U}_{\mathrm{ctrl}}(x) \triangleq \mathsf{Proj}_u\left(\widehat{\mathcal{I}}_{n(x)}, x\right) \tag{19}$$

in other words, $\mathcal{U}_{\mathrm{ctrl}}(x)$ is the set of constant control values which are guaranteed to keep the trajectory from x inside $\mathcal{S}$ and lead it into $\mathcal{K}_{n(x)-1}$ at the next sample time for any sample period in the range $[\delta^{\mathrm{F}}, \delta^{\mathrm{F}} + \delta^{\mathrm{J}}]$ and for any disturbance signal. These are the control values which permit x to be part of $\mathcal{K}_{n(x)}$.

2.3 Algorithm Output Properties

Before showing the relationship between (16) and the kernel (6), we characterize the location of trajectories at the sample times.

LEMMA 1. *For any sample time sequence $\mathcal{T}$ if $x(t_k) \in \mathsf{Disc}_j(\mathcal{S})$ for some $j \in 1, \ldots, \bar{N}$, then there exists a constant $u_k \in \mathcal{U}$ such that*

$$x(t_{k+1}) \in \begin{cases} \mathcal{S}, & \text{if } j = 1; \\ \mathsf{Disc}_{j-1}(\mathcal{S}) & \text{if } 2 \le j \le \bar{N}; \end{cases}$$

for any disturbance input $v(\cdot)$.

PROOF. Let $t_{k+1} - t_k = \delta^{\mathrm{F}} + \delta^{\mathrm{J}}_k$. By (16), there exists $u_k \in \mathcal{U}$ such that $\begin{bmatrix} x(t_k) & u_k \end{bmatrix}^T \in \mathcal{I}_j$. We consider the two cases separately:

If $j = 1$: By (10) and (12), for any $v(\cdot)$ and $\delta_k \in [0, \delta^{\mathrm{F}} + \delta^{\mathrm{J}}]$, $\begin{bmatrix} x(t_k + \delta_k) & u_k \end{bmatrix}^T \in \mathcal{S} \times \mathcal{U}$, which implies that $x(t_k + \delta^{\mathrm{F}} + \delta^{\mathrm{J}}_k) = x(t_{k+1}) \in \mathcal{S}$.

If $2 \le j \le \bar{N}$: By (10) and (14), for any $v(\cdot)$ and $\delta^{\mathrm{J}}_k \in [0, \delta^{\mathrm{J}}]$, we have that $\begin{bmatrix} x(t_k + \delta^{\mathrm{J}}_k) & u_k \end{bmatrix}^T \in \mathcal{R}_j$. By (11) and (13), $\begin{bmatrix} x(t_k + \delta^{\mathrm{J}}_k + \delta^{\mathrm{F}}) & u_k \end{bmatrix}^T \in \mathsf{Disc}_{j-1}(\mathcal{S}) \times \mathcal{U}$, which implies that $x(t_k + \delta^{\mathrm{J}}_k + \delta^{\mathrm{F}}) = x(t_{k+1}) \in \mathsf{Disc}_{j-1}(\mathcal{S})$. □

With this characterization in place, we can map out the relationship between the algorithm's representation (16) and the true kernel (6):

PROPOSITION 2. *The (jitter robust) sampled data discriminating kernel is given by*

$$\mathsf{Disc}_{\bar{N}}(\mathcal{S}) \subseteq \mathsf{Disc}_{\mathsf{sd}}([0, T], \mathcal{S}).$$

PROOF. We seek to show

$$x_0 \in \mathsf{Disc}_{\bar{N}}(\mathcal{S}) \implies x_0 \in \mathsf{Disc}_{\mathsf{sd}}([0, T], \mathcal{S}).$$

Assume that $x_0 \in \mathsf{Disc}_{\bar{N}}(\mathcal{S})$. We first show that for any $\mathcal{T}$ and $v(\cdot)$, $x(t_k) \in \mathsf{Disc}_{\bar{N}-k}(\mathcal{S})$ for $k = 0, 1, \ldots, \bar{N} - 1$ by induction. The base case $x(t_0) = x_0 \in \mathsf{Disc}_{\bar{N}}(\mathcal{S})$ is true by assumption. For the inductive step, if $x(t_k) \in \mathsf{Disc}_{\bar{N}-k}(\mathcal{S})$ then by lemma 1 there exists a constant input $u_k \in \mathcal{U}$ such that $x(t_{k+1}) \in \mathsf{Disc}_{\bar{N}-k-1}(\mathcal{S})$ for any $v(\cdot)$.

By (16), for that same $u_k \in \mathcal{U}$, $\begin{bmatrix} x(t_k) & u_k \end{bmatrix}^T \in \mathcal{I}_1$. By (12), for any $t \in [t_k, t_k + \delta^{\mathrm{F}} + \delta^{\mathrm{J}}]$, $\begin{bmatrix} x(t) & u_k \end{bmatrix}^T \in \mathcal{S} \times \mathcal{U}$ for any $v(\cdot)$, which implies that for any $\mathcal{T}$ and $v(\cdot)$, $x(t) \in \mathcal{S}$ for any $t \in [t_k, t_{k+1}]$, where t_k and t_{k+1} are consecutive elements of $\mathcal{T}$. Since

$$\bigcup_{k=0}^{\bar{N}-1} [t_k, t_{k+1}] = [t_0, t_{\bar{N}}] \supseteq [0, T]$$

we have shown that $x(t) \in \mathcal{S}$ for all $t \in [0, T]$ for any $\mathcal{T}$ and $v(\cdot)$, and hence that $x_0 \in \mathsf{Disc}_{\mathsf{sd}}([0, T], \mathcal{S})$. □

Intuitively, the substitution of the reach set (13) into the algorithm seems dubious because it allows the trajectory to be outside $\mathsf{Disc}_{j-1}(\mathcal{S})$ during the fixed portion of the sampling period as long as it is inside before the end. However, the constraint we need to satisfy is that the trajectory remains in $\mathcal{S}$ for all times, not $\mathsf{Disc}_{j-1}(\mathcal{S})$. By (15) and (16), only states in $\mathsf{Proj}_x(\mathcal{I}_1)$ are under consideration; consequentially, even though the trajectory may leave $\mathsf{Disc}_{j-1}(\mathcal{S})$ during the sample period, by (12) it does not leave $\mathcal{S}$.

Proposition 2 shows that the improved algorithm's results remain conservative. For systems without jitter ($\delta^{\mathrm{J}} = 0$) it is straightforward to show that the improved algorithm's $\mathsf{Disc}_j(\mathcal{S})$ is always a subset of the corresponding kernel approximation generated by the algorithm in [17] and is hence at least as accurate; however, we can in fact show that the improved algorithm is tight in this case.

PROPOSITION 3. *If $\delta^J = 0$ and $T = N\delta^F$ for some integer $N > 0$ then*

$$\mathsf{Disc}_N(\mathcal{S}) = \mathsf{Disc}_{\mathsf{sd}}([0, T], \mathcal{S}).$$

PROOF. We only need to show

$$x_0 \in \mathsf{Disc}_{\mathsf{sd}}([0, T], \mathcal{S}) \implies x_0 \in \mathsf{Disc}_N(\mathcal{S})$$

because the converse implication has already been proven in proposition 2. Assume $x_0 \in \mathsf{Disc}_{\mathsf{sd}}([0, T], \mathcal{S})$ and observe that if $\delta^{\mathrm{J}} = 0$ and $T = N\delta^{\mathrm{F}}$ then $\mathcal{T}$ is a singleton containing only the time sample sequence $t_k = k\delta^{\mathrm{F}}$ for $k = 0, 1, \ldots, N$. By (6) there exists a $u_{\mathrm{pw}}(\cdot)$ such that for all $v(\cdot)$ and $t \in [0, T]$, $x(t) \in \mathcal{S}$. Let $u_k = u_{\mathrm{pw}}(t_k) \in \mathcal{U}$. Because $x(t) \in \mathcal{S}$ for all $v(\cdot)$ and $t \in [t_k, t_{k+1}]$ for $k = 0, 1, \ldots, N - 1$, by (10) and (12)

$$\begin{bmatrix} x(t_k) \\ u_k \end{bmatrix} \in \mathcal{I}_1. \tag{20}$$

We proceed by induction to show that $x(t_k) \in \mathsf{Disc}_{N-k}(\mathcal{S})$ for $k = N-1, N-2, \ldots, 0$. By (8), (16) and (20), we have the base case $x(t_{N-1}) \in \mathsf{Disc}_1(\mathcal{S})$. Now assume that $x(t_{k+1}) \in \mathsf{Disc}_{N-k-1}(\mathcal{S})$. By (14) and (13), $\begin{bmatrix} x(t_k) & u_k \end{bmatrix}^T \in \mathcal{I}_{N-k} = \mathcal{R}_{N-k}$, which by (8), (16) and (20) implies that $x(t_k) \in \mathsf{Disc}_{N-k}(\mathcal{S})$. In particular, $x(t_0) = x_0 \in \mathsf{Disc}_N(\mathcal{S})$. □

Conservativeness of the approximation $\mathsf{Disc}_k(\mathcal{S})$ was the key property of the algorithm from [17] which made it possible to synthesize a control policy which guaranteed finite horizon satisfaction of constraint $\mathcal{K}_0$. Because the improved algorithm maintains this conservativeness, it is straightforward to show a similar result.

THEOREM 4. *Let trajectory $x(\cdot)$ solve (4)–(5) with initial condition $x(0) = x_0$ and sampled feedback control policy $u_{fb}(x) \in \mathcal{U}_{ctrl}(x)$ given by (19). If $x_0 \in \mathcal{K}_0$ then $x(t) \in \mathcal{K}_0$ for all sample time sequences $\mathcal{T} \in \mathbb{T}$, all disturbance inputs $v(\cdot)$ and all $t \in [0, n(x_0)\delta^F]$.*

PROOF. A straightforward modification of the proof of [17, Theorem 4] to use the improved approximation (16). □

2.4 Improved Algorithm Demonstration

We demonstrate the two key improvements to the algorithm—tightness and jitter robustness—through two examples. Both examples are implemented using an HJ PDE formulation; see [17, section 5] for the details. The only new operator required by the improved algorithm is the reach set (11), which is approximated using the same machinery as the invariance kernel (10) except that the HJ PDE for the reach set

$$D_t\phi + \max H(\tilde{x}, D_{\tilde{x}}\phi) = 0$$

omits the "max with zero" constraint on the sign of the time derivative which ensures that the invariance kernel only shrinks. Otherwise, the rest of the HJ PDE for the reach set is the same as that for the invariance kernel in [17, equation (18)]. Because both the invariance kernel and reach set solve essentially the same PDE, the execution time of the new algorithm using the HJ formulation is essentially unchanged from that of the old algorithm. We use the Toolbox of Level Set Methods [18] to approximate the solution to the HJ PDEs.

As a demonstration of proposition 3, we revisit the example from [17, section 4.3] which was used to prove that the

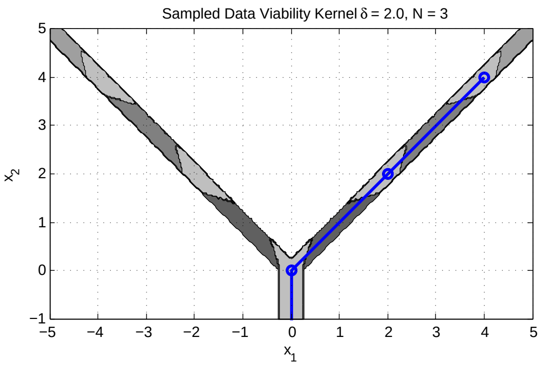

Figure 1: A demonstration of proposition 3. In this case $\mathcal{S}$ is the Y-shaped shaded region. The states in $\mathsf{Disc}_j(\mathcal{S})$ for $j = 0, 1, 2, 3$ are shown darkest to lightest (darker colored sets also contain all lighter colored states). The solid blue line shows a trajectory starting from the upper right which remains in $\mathcal{S}$ for three sample periods; the input for this trajectory is sampled at the points marked by small circles. Using the improved algorithm, the starting point for this trajectory is correctly identified as being inside $\mathsf{Disc}_3(\mathcal{S})$. In fact, all of the lightest shaded regions in this figure are part of $\mathsf{Disc}_\infty(\mathcal{S})$, but the computation of $\mathsf{Disc}_j(\mathcal{S})$ is only performed for $j \leq 3$.

old algorithm was strictly conservative. This example has no jitter ($\delta^{\mathrm{J}} = 0$) and no disturbance input v; consequently, we are computing a viability kernel instead of a discriminating kernel. Let $f(x, u, v) = \begin{bmatrix} u & -1 \end{bmatrix}^T$ in (4) with $\mathcal{U} = [-1, +1]$. Let $\mathcal{S}$ be the Y-shaped shaded region shown in figure 1 (the arms and leg of the Y are assumed to extend outward to infinity). The upper arms of the Y have constant width and a 45° slope. The vertical leg of $\mathcal{S}$ is viable for all $\delta^{\mathrm{F}} > 0$, but for $\delta^{\mathrm{F}} = 2$ there are regions of the upper arms which give rise to sampled data trajectories which inevitably leave $\mathcal{S}$ because no sample points occur where the upper arms join and hence the input cannot be switched in time from the $u = \pm 1$ value required to make the upper arms viable to the $u = 0$ value required to make the lower leg viable. On the other hand, there are states along the upper arms which give rise to trajectories which remain viable for all time; for example, the trajectory shown in figure 1 starts at $\begin{bmatrix} +4 & +4 \end{bmatrix}^T$ and uses input signal

$$u_{\mathrm{pw}}(t) = \begin{cases} -1 & 0 \leq t < 4; \\ 0 & t \geq 4. \end{cases}$$

As shown in [17, figures 2 and 3], the old algorithm failed to detect these patches of viability in the upper arms, while figure 1 demonstrates that the improved algorithm captures both the viable and non-viable states in the arms correctly.

As a demonstration of proposition 2 for systems with sampling jitter we consider a similar artificial example with no disturbance input v. Let $f(x, u, v) = u$ in (4) and let

$$u_{\mathrm{up}} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \qquad u_{\mathrm{right}} = \begin{bmatrix} 1 \\ 0 \end{bmatrix};$$

$$\mathcal{U} = \{u \mid u = \lambda u_{\mathrm{up}} + (1 - \lambda) u_{\mathrm{right}} \text{ for } 0 \leq \lambda \leq 1\}.$$

The dynamics can be summarized as "trajectories can move up and/or to the right." We choose the shaded region shown in figure 2 as set $\mathcal{S}$ (the set is assumed to extend horizontally outward to infinity). The width of the vertical chimney of $\mathcal{S}$ is 0.5, the height of the two horizontal ducts is 0.2, and the distance between the bottom (or top) edges of $\mathcal{S}$ on the left and right sides is 1. We fix the maximum sample period $\delta^{\mathrm{F}} + \delta^{\mathrm{J}} = 1$ and number of samples $\bar{N} = 5$, and then explore the effect of changing the maximum jitter. The rightmost horizontal duct of $\mathcal{S}$ is in $\mathsf{Disc}_5(\mathcal{S})$ (the lightest shading) for any jitter (it is actually in $\mathsf{Disc}_\infty(\mathcal{S})$), but in the same manner as the upper arms of the constraint set in the previous example, the viability of states in the vertical chimney or leftmost horizontal duct of $\mathcal{S}$ depends on whether subsequent sampling times will occur where the input value needs to be changed when switching from the leftmost horizontal duct into the vertical chimney or from the vertical chimney into the rightmost horizontal duct.

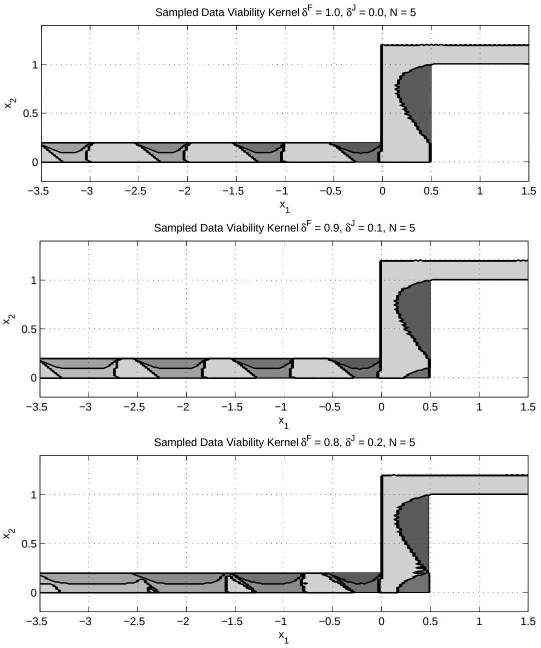

Figure 2: A demonstration of proposition 2. Each plot shows $\mathsf{Disc}_j(\mathcal{S})$ for $j = 0, 1, \ldots, 5$ (from darkest to lightest shading). The top plot shows $\delta^{\mathbf{F}} = 1.0$ and $\delta^{\mathbf{J}} = 0.0$, the middle plot shows $\delta^{\mathbf{F}} = 0.9$ and $\delta^{\mathbf{J}} = 0.1$, and the bottom plot shows $\delta^{\mathbf{F}} = 0.8$ and $\delta^{\mathbf{J}} = 0.2$. Note how the cumulative effect of timing jitter rapidly removes states in the leftmost horizontal duct from $\mathsf{Disc}_5(\mathcal{S})$.

The uppermost plot in figure 2 shows the case $\delta^{\mathrm{J}} = 0$. The vertical chimney is mostly in $\mathsf{Disc}_5(\mathcal{S})$ except for a patch in the upper right that is not even in $\mathsf{Disc}_1(\mathcal{S})$: trajectories arising from these states cannot round the corner into the rightmost horizontal duct and hence cannot stay viable for even one sample period[1]. The leftmost horizontal duct

[1] The roughness of the left edge of this patch (and in general the boundary of other patches) is a symptom of relatively coarse sampling of $\mathcal{U}$: The discretization of $\Omega \times \mathbb{U}$ was $201 \times 161 \times 25$. A much better approximation can be easily achieved by a finer discretization in the input dimension at the cost of linearly increased computation time.

clearly shows patches in $\mathsf{Disc}_5(\mathcal{S})$ which are all the same width (0.5) as the vertical chimney plus a triangular patch at the front where it is possible to cut the lower left corner of $\mathcal{S}$ when entering the vertical chimney. The other patches in this horizontal duct consists of two parts: one darker and one lighter. The darker states along the top edge of the duct are those which give rise to trajectories which are unable to round the first corner into the vertical chimney and hence fall outside of $\mathsf{Disc}_j(\mathcal{S})$ for $j = 1, 2, 3, 4$ (from right to left), while the lighter states along the bottom edge are those whose trajectories can round the first corner but land in the non-viable patch in the chimney, cannot round the second corner into the upper horizontal duct and hence are inside $\mathsf{Disc}_j(\mathcal{S})$ but outside $\mathsf{Disc}_{j+1}(\mathcal{S})$ for $j = 1, 2, 3, 4$ (from right to left).

The middle plot in figure 2 shows the case $\delta^{\mathrm{J}} = 0.1$. Despite the fact that the maximum sample period is unchanged and the jitter can only result in more frequent sampling and opportunities to modify the input, $\mathsf{Disc}_5(\mathcal{S})$ has shrunk. The same pattern of $\mathsf{Disc}_j(\mathcal{S})$ for $j < 5$ appears, with the addition of a small patch at the bottom right of the chimney that lies in $\mathsf{Disc}_1(\mathcal{S})$ from which trajectories can stay in the chimney for one sample period but cannot always round the upper corner into the rightmost horizontal duct because of the possibility of an early sample time. More importantly, the patches in the leftmost horizontal duct that lie within $\mathsf{Disc}_5(\mathcal{S})$ are smaller: the first has width 0.4 at the top, the second width 0.3, and so on. This shrinking width is due to the compounding of jitter over multiple sample periods.

The bottom plot in figure 2 shows the case $\delta^{\mathrm{J}} = 0.2$. Because of the compounding effect of jitter, almost none of the leftmost horizontal duct lies in $\mathsf{Disc}_5(\mathcal{S})$ anymore.

3. QUADROTOR ALTITUDE MAINTENANCE

In this section we use a safety problem for a quadrotor to demonstrate a scalable implementation of the improved algorithm for systems with LTI dynamics using an ellipsoidal representation of the various sets and kernels.

3.1 Modeling and Constraints

Consider the problem of altitude control for the autonomous quadrotor testbed modeled in [1]:

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= k_T x_3^2 - g, \\ \dot{x}_3 &= k_p(u - x_3); \end{aligned} \tag{21}$$

where x_1 is the vertical position of the vehicle, x_2 is the vertical velocity, and x_3 is the current average angular acceleration (related to thrust) while the input $u \in \mathbb{R}$ is the commanded average angular acceleration. The actuator response to input commands is modeled as a first order system with a time constant of $1/k_p$ to account for inherent rotor delay. The effect of x_3 on the acceleration of the vehicle is modeled as a quadratic function with a normalizing coefficient k_T. Values of $k_p = 6.6667$ and $k_T = 0.1222$ were empirically identified. The constant $g = 9.8$ is the gravitational acceleration.

The input is constrained as $0 \leq u \leq 10$ (measured in counts). The position and velocity constraints are chosen based on the ceiling height of the room and the physical characteristics of the quadrotor: $0.5\,\mathrm{m} \leq x_1 \leq 2.8\,\mathrm{m}$ and $-1.5\,\mathrm{m/s} \leq x_2 \leq 1.5\,\mathrm{m/s}$. The bounds on x_3 are discussed when we linearize the dynamics below.

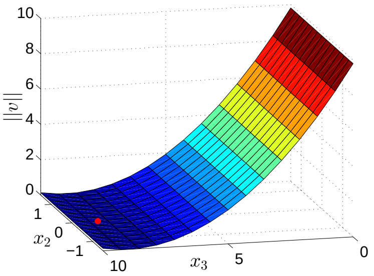

Figure 3: For fixed x_1 and u, the perturbation magnitude to x_2 depends on x_3. In the neighborhood of the hover x_{eq} (shown as a red dot) the disturbance is small, while away from this linearization point the unmodeled nonlinearities are magnified.

For the purposes of this example we assume that control commands can only be modified every $\delta = 100\,\mathrm{ms}$; the corresponding 10 Hz update cycle is somewhat low but within an order of magnitude of the rate at which typical quadrotors read sensors and update actuators. We also assume no timing jitter so that $\delta^{\mathrm{F}} = \delta$ and $\delta^{\mathrm{J}} = 0$. We wish to analyze the safety of this quadrotor as it moves vertically in the room, and synthesize the set-valued safety-preserving state-feedback control that ensures the vehicle does not crash into the floor or the ceiling over the time interval $[0, 1]$.

The scalable ellipsoidal representation requires LTI dynamics, so we linearize (21) about a hover condition at state

$$x_{\mathrm{eq}} = \begin{bmatrix} 2 & 0 & 8.96 \end{bmatrix}^T; \tag{22}$$

in other words, at an altitude of 2 m using the empirically measured input of $u_{\mathrm{eq}} = 8.96$ counts. In order to ensure that our results are safe despite this linearization, we take any linearization error into account through the disturbance input. The linearized model is

$$\dot{\bar{x}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 2(8.96)k_T \\ 0 & 0 & -k_p \end{bmatrix} \bar{x} + \begin{bmatrix} 0 \\ 0 \\ k_p \end{bmatrix} \bar{u} + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} v, \tag{23}$$

where $\bar{x} \triangleq x - x_{\mathrm{eq}}$, $\bar{u} \triangleq u - u_{\mathrm{eq}}$, and $v \in \mathbb{R}$ is the disturbance. The bounds on v depend on the range of x_3: Looser bounds on x_3 require a larger set of possible disturbances to account for more pronounced nonlinearities in the velocity dynamics (see figure 3). Here we choose the constraint $x_3 \in [8, 10]$, which yields a worst-case linearization error of 0.1. Since the error dynamics force x_2 to move in the positive direction, the disturbance v is constrained to the set $\mathcal{V} = [0, 0.1]$.

3.2 A Scalable Algorithm for Safety Analysis

In [17] we described an implementation of that paper's abstract algorithm using ellipsoidal techniques from reachability [11, 12, 13] that scale polynomially with state space dimension. In this section we describe two modifications that were made to analyze the quadrotor example. First, we use a slightly different semi-definite program (SDP) to find the "best" inscribed ellipsoid when implementing the intersection operator. Second, we use the improved sampled data discriminating kernel algorithm from section 2.2.

In this implementation, sets are represented by ellipsoids. An ellipsoid in $\mathbb{R}^d$ is defined by

$$\begin{aligned}\mathcal{E}(q, \mathrm{H}) &\triangleq \{\mathrm{H}y + q \in \mathbb{R}^d \mid \|y\|_2 \leq 1\} \\ &= \{y \in \mathbb{R}^d \mid (y-q)^T \mathrm{H}^{-2}(y-q) \leq 1\},\end{aligned}$$

where $q \in \mathbb{R}^d$ is the center, $\mathrm{H} = \mathrm{H}^T \in \mathbb{R}^{d \times d}$, and $\mathrm{H}\mathrm{H}^T = \mathrm{H}^2$ is the symmetric positive definite shape matrix. For matrix A, the linear mapping of an ellipsoid is also an ellipsoid

$$\mathrm{A}\mathcal{E}(q, \mathrm{H}) = \mathcal{E}(\mathrm{A}q, \mathrm{AH})$$

We call a finite union of ellipsoids a piecewise ellipsoidal set.

3.2.1 Maximum Trace Inscribed Ellipsoids

A key operation in the sampled data discriminating kernel algorithm is set intersection. It is well known that the intersection of nonempty ellipsoids $\{\mathcal{Y}_i\}$ is not an ellipsoid but that an ellipsoidal underapproximation of that intersection can be found through one of several SDPs. In [17] we sought a maximum volume underapproximation (using a log det objective function in the SDP), but here we switch to a maximum trace underapproximation (using a trace objective function in the SDP) because the latter is more easily adapted to use separate scaling factors for the dimensions in Ω and $\mathbb{U}$.

A further twist in the intersection issue arises because the intersection is performed (16) in the space $\Omega \times \mathbb{U}$ but is then projected into Ω via (8) or $\mathbb{U}$ via (19) and (9). For that reason it is not immediately clear how to define the "best" underapproximating ellipsoid. In [17] we only demonstrated the case using the maximum volume ellipsoid in $\Omega \times \mathbb{U}$, but here we consider a spectrum of possibilities.

To explore different projections of ellipsoids, let $\mathrm{P} \in \mathbb{R}^{d_1 \times d_2}$ with $d_1 \leq d_2$ be a matrix such that $\mathrm{P}^T\mathrm{P}$ is a projection matrix (so $(\mathrm{P}^T\mathrm{P})^2 = \mathrm{P}^T\mathrm{P}$). In particular, we will use matrices

$$\mathrm{P}_x = \begin{bmatrix}\mathrm{I}_{d_x} & 0_{d_x \times d_u}\end{bmatrix} \qquad \text{and} \qquad \mathrm{P}_u = \begin{bmatrix}0_{d_u \times d_x} & \mathrm{I}_{d_u}\end{bmatrix}$$

where $I_d \in \mathbb{R}^{d \times d}$ is an identity matrix and $0_{d_1 \times d_2} \in \mathbb{R}^{d_1 \times d_2}$ is a zero matrix. Given an augmented state $\tilde{x} = \begin{bmatrix}x & u\end{bmatrix}^T$, we then have that $\mathrm{P}_x\tilde{x} = x$ and $\mathrm{P}_u\tilde{x} = u$.

A maximum trace inscribed ellipsoid is the ellipsoid with maximum semi-axis lengths lying within a given constraint. In particular, we will use the maximum trace inscribed ellipsoid $\mathcal{E}_{\cap_i \mathcal{Y}_i}$ to underapproximate the intersection of nonempty ellipsoids $\{\mathcal{Y}_i\}$. It can be determined by solving a convex semi-definite program. We slightly extend the technique to allow sets $\mathcal{Y}_i$ which can be either an ellipsoid $\mathcal{Y}_i = \mathcal{E}(q_i, \mathrm{H}_i)$ or the tensor product of lower dimensional ellipsoids

$$\begin{aligned}\mathcal{Y}_i &= \mathcal{Y}_{i,x} \times \mathcal{Y}_{i,u} \\ \text{where } \mathcal{Y}_{i,x} &\triangleq \mathcal{E}(q_{i,x}, \mathrm{H}_{i,x}) \subset \mathbb{R}^{d_x} \\ \text{and } \mathcal{Y}_{i,u} &\triangleq \mathcal{E}(q_{i,u}, \mathrm{H}_{i,u}) \subset \mathbb{R}^{d_u}.\end{aligned}$$

For notational simplicity we have assumed that the lower dimensional ellipsoids happen to be in the x and u subspaces of the augmented state space $\tilde{x}$, although the formulation can easily be generalized.

We will also modify the objective of the optimization and introduce a tradeoff factor $\alpha \in [0, 1]$ that weights the objective somewhere between finding an inscribed ellipsoid whose projection has maximal trace in Ω and one that has maximal trace in $\mathbb{U}$. A value of $\alpha = 0.5$ will recover the ellipsoid which has maximal trace in $\Omega \times \mathbb{U}$.

If $\cap_i \mathcal{Y}_i \neq \emptyset$, solve the semidefinite program (SDP)

$$\begin{gathered}\text{minimize} - (1-\alpha)\operatorname{Tr}(\mathrm{P}_x\bar{\mathrm{H}}\mathrm{P}_x^T) - \alpha\operatorname{Tr}(\mathrm{P}_u\bar{\mathrm{H}}\mathrm{P}_u^T) \\ \text{over } \bar{\mathrm{H}} \in \mathbb{R}^{d \times d},\ \bar{q} \in \mathbb{R}^d, \text{ and } \lambda_i \in \mathbb{R}\end{gathered} \tag{24}$$

subject to constraints for $i = 1, 2, \ldots$ either of the form

$$\begin{gathered}\lambda_i > 0 \\ \begin{bmatrix}1-\lambda_i & 0 & (\bar{q}-q_i)^T \\ 0 & \lambda_i\mathrm{I} & \bar{\mathrm{H}} \\ (\bar{q}-q_i) & \bar{\mathrm{H}} & \mathrm{H}_i^2\end{bmatrix} \geq 0,\end{gathered} \tag{25}$$

if $\mathcal{Y}_i = \mathcal{E}(q_i, \mathrm{H}_i)$ or of the form

$$\begin{gathered}\lambda_{i,x} > 0 \\ \lambda_{i,u} > 0 \\ \begin{bmatrix}1-\lambda_{i,x} & 0 & (\mathrm{P}_x\bar{q}-q_{i,x})^T \\ 0 & \lambda_{i,x}\mathrm{I} & \mathrm{P}_x\bar{\mathrm{H}}\mathrm{P}_x^T \\ (\mathrm{P}_x\bar{q}-q_{i,x}) & \mathrm{P}_x\bar{\mathrm{H}}\mathrm{P}_x^T & \mathrm{H}_{i,x}^2\end{bmatrix} \geq 0 \\ \begin{bmatrix}1-\lambda_{i,u} & 0 & (\mathrm{P}_u\bar{q}-q_{i,u})^T \\ 0 & \lambda_{i,u}\mathrm{I} & \mathrm{P}_u\bar{\mathrm{H}}\mathrm{P}_u^T \\ (\mathrm{P}_u\bar{q}-q_{i,u}) & \mathrm{P}_u\bar{\mathrm{H}}\mathrm{P}_u^T & \mathrm{H}_{i,u}^2\end{bmatrix} \geq 0\end{gathered} \tag{26}$$

if $\mathcal{Y}_i = \mathcal{Y}_{i,x} \times \mathcal{Y}_{i,u}$, where I and 0 are appropriately sized identity and zero matrices. The optimal values $\bar{\mathrm{H}}^*$ and $\bar{q}^*$ define the inscribed ellipsoid for the chosen tradeoff factor α:

$$\mathsf{Inscribed}_\alpha\left(\cap_i \mathcal{Y}_i\right) \triangleq \mathcal{E}\left(\bar{q}^*, \bar{\mathrm{H}}^*\right).$$

3.2.2 The Improved Piecewise Ellipsoidal Algorithm

When a set $\mathcal{S}$ is not an ellipsoid, its ellipsoidal approximation is denoted by $\mathcal{E}_\mathcal{S}$. In this example the sets $\mathcal{U}$ and $\mathcal{V}$ are already ellipsoids as they are both intervals in $\mathbb{R}$. However, sets of the form $\mathcal{S} \times \mathcal{U}$ in (12) and (13) are the tensor product of two ellipsoids, and hence are very poorly approximated by a single inscribed ellipsoid. Instead, we *over-approximate* $\mathcal{S} \times \mathcal{U}$ with an ellipsoid whose projection into Ω is tight to $\mathcal{S}$ but whose projection into $\mathbb{U}$ is much larger than $\mathcal{U}$ (but still bounded). Call this set $\mathcal{E}_{\mathcal{S} \times \mathbb{U}}$. This overapproximation can be much tighter to $\mathcal{S} \times \mathcal{U}$ for $u \in \mathcal{U}$, although it does allow for $u \notin \mathcal{U}$. Fortunately, the augmented dynamics are zero in the u dimensions, so we can remove these extraneous augmented states through a later intersection with $\mathcal{E}_{\Omega \times \mathcal{U}}$, which is an ellipsoid whose projection into $\mathbb{U}$ is tight to $\mathcal{U}$ but whose projection into Ω is much larger than $\mathcal{S}$ (but still bounded).

As in [17], we use the algorithm from [8, 10, 9] to implement an ellipsoidal underapproximation $\mathcal{E}_{\mathsf{Inv}(\cdot,\cdot)}$ of invariance kernels (10) through a sequence of reach sets and intersections. Unlike [17], the inscribed ellipsoid calculated during each intersection is chosen using the maximum trace objective (24) with a prespecified tradeoff factor α. Ellipsoidal underapproximations $\mathcal{E}_{\mathsf{Reach}(\cdot,\cdot)}$ of robust reach sets (11) can be implemented directly with the algorithms from [12, 13]. Both of these algorithms generate a piecewise ellipsoidal underapproximation of the desired set parameterized by a finite collection of direction vectors ℓ.

The key steps of the improved sampled data discriminating kernel algorithm (12)–(16) for a given direction vector ℓ

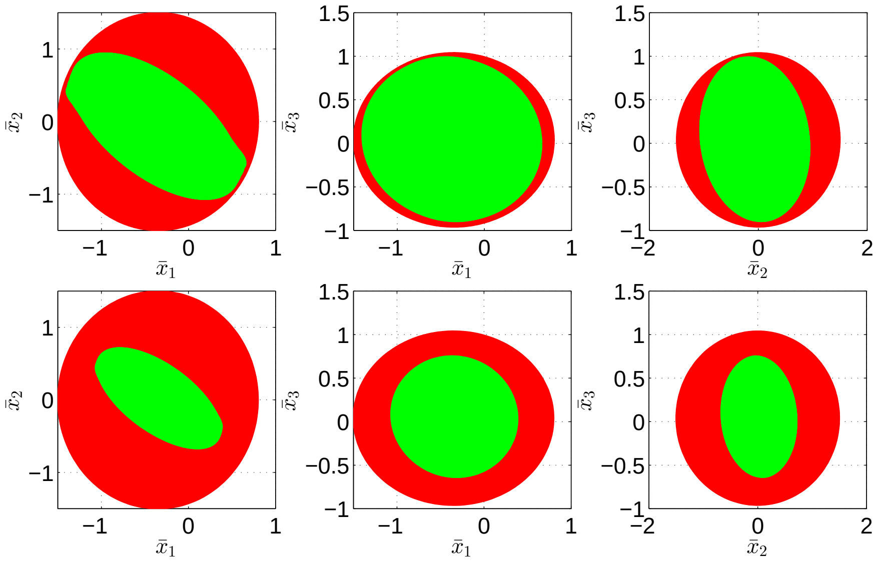

Figure 4: A piecewise ellipsoidal under-approximation of the sampled-data discriminating kernel (green) for the quadrotor example using (i) the improved algorithm (top row), and (ii) the algorithm presented in [17] (bottom row). For each case 20 terminal directions ℓ were chosen, of which 15 and 13, respectively, resulted in nonempty ellipsoids. The state constraint is also shown (red).

are then

$$\begin{aligned}
\mathcal{E}_{\mathcal{I}_1}(\ell) &= \mathcal{E}_{\mathsf{Inv}\left([0,\delta^{\mathrm{F}}],\mathcal{S}\times\mathbb{U}\right)}(\ell), \\
\mathcal{E}_{\mathcal{R}_j}(\ell) &= \mathsf{Reach}\left([0,\delta^{\mathrm{F}}],\mathcal{E}_{\mathsf{Disc}_{j-1}(\mathcal{S})\times\mathbb{U}}(\ell)\right), \\
\mathcal{E}_{\widehat{\mathcal{I}}_j}(\ell) &= \mathsf{Inscribed}_\alpha\left(\mathcal{E}_{\mathcal{I}_1}(\ell)\cap\mathcal{E}_{\mathcal{R}_j}(\ell)\right), \\
\mathcal{E}_{\mathsf{Disc}_1(\mathcal{S})}(\ell) &= \mathsf{Proj}_x\left(\mathsf{Inscribed}_0\left(\mathcal{E}_{\mathcal{I}_1}(\ell)\cap\mathcal{E}_{\Omega\times\mathcal{U}}\right)\right), \\
\mathcal{E}_{\mathsf{Disc}_j(\mathcal{S})}(\ell) &= \mathsf{Proj}_x\left(\mathsf{Inscribed}_0\left(\mathcal{E}_{\widehat{\mathcal{I}}_j}(\ell)\cap\mathcal{E}_{\Omega\times\mathcal{U}}\right)\right),
\end{aligned}$$

for $j = 2, 3, \ldots$. We use a tradeoff factor of α when computing $\mathcal{E}_{\widehat{\mathcal{I}}_j}$ in order to keep the resulting inscribed ellipsoid from collapsing in $\mathbb{U}$ and hence producing an empty $\mathcal{U}_{\mathrm{ctrl}}$ (defined below), but we can safety use a tradeoff factor of 0 when computing $\mathcal{E}_{\mathsf{Disc}_j(\mathcal{S})}$ because we immediately project the resulting inscribed ellipsoid into Ω and hence its extent in $\mathbb{U}$ is irrelevant.

For the safety problem, we underapproximate $\mathcal{K}_j$ with the ellipsoid $\mathcal{E}_{\mathsf{Disc}_j(\mathcal{K}_0)}$, define $n(x)$ as in (18), and create a safe control policy

$$\mathcal{U}_{\mathrm{ctrl}}(x) = \mathsf{Proj}_u\left(\mathcal{E}_{\widehat{\mathcal{I}}_{n(x)}}, x\right) \cap \mathcal{U}. \qquad (27)$$

It is necessary to clip the projection of $E_{\widehat{\mathcal{I}}_{n(x)}}$ by intersection with $\mathcal{U}$ because $\mathcal{E}_{\mathcal{R}_j}$ was computed starting from $\mathcal{E}_{\mathsf{Disc}_{j-1}(\mathcal{S})\times\mathbb{U}}$ rather than $\mathcal{E}_{\mathsf{Disc}_{j-1}(\mathcal{S})\times\mathcal{U}}$.

3.3 Results

We implement the ellipsoidal operators using the Ellipsoidal Toolbox (ET) [14] version 1.1.3 and the semi-definite programs using YALMIP [15] on an Intel Core i7-3520M at 2.9 GHz with 16 GB RAM running 64-bit Windows 7 Pro and Matlab R2011b.

Applying the above algorithm to the quadrotor altitude maintenance problem yields a significant improvement over the results generated by our previous method [17]. Figure 4 compares the approximations $\cup_\ell \mathcal{E}_{\mathsf{Disc}_N(\mathcal{S})}(\ell)$ of the sampled-data discriminating kernel for this problem generated using both algorithms, and it is clear that the result generated by the algorithm described in this paper is significantly larger. There are several reasons for the improved results:

- The improved abstract algorithm of section 2 is tight in this jitter-free case, while the algorithm in [17] was conservative but not tight.
- The improved piecewise ellipsoidal algorithm does not attempt to approximate the tensor product set $\mathcal{S} \times \mathcal{U}$ with a single inscribed ellipsoid, but rather uses two ellipsoids $\mathcal{S} \times \mathbb{U}$ and $\Omega \times \mathcal{U}$ (as discussed at the beginning of section 3.2.2) in a manner which is conservative but much closer to the true tensor product.
- The invariance kernel operator—which is implemented with alternating reach sets and inscribed approximations of intersections—is used only during the first sample period in this algorithm. Subsequent sample periods are implemented using just a single reach set and inscribed approximation of an intersection.

In addition to the improvement in accuracy, the new algorithm is also dramatically faster: The sets shown in the top row of Figure 4 were generated in just 5 min, while those in the bottom row required 173 min[2].

Given $\bar{x} \in \cup_\ell \mathcal{E}_{\mathsf{Disc}_N(\mathcal{S})}(\ell)$, there exists a set of control inputs $\mathcal{U}_{\mathrm{ctrl}}(\bar{x}) \subseteq \mathcal{U}$ from which the quadrotor can choose a value and remain within the state constraint over at least the $[0, 1]$ time horizon. Figure 5 shows slices of $\mathcal{E}_{\widehat{\mathcal{I}}_{\bar{N}}}$ (from which $\mathcal{U}_{\mathrm{ctrl}}$ is constructed via (27)) for different values of the state $\bar{x}$ when the discriminating kernel approximation is computed for just one direction, $\ell = \begin{bmatrix}0.5 & 0.5 & 0.5 & 0.5\end{bmatrix}^T$. Each subplot corresponds to a slice in $\bar{x}_3$ and shows three slices in $\bar{x}_2$: one through the center of $\mathcal{E}_{\widehat{\mathcal{I}}_{\bar{N}}}$ and two through

[2] The larger number is likely exaggerated due to ET (or YALMIP) being repeatedly called in a for-loop, which in our experience yields artificial slowing of the whole process.

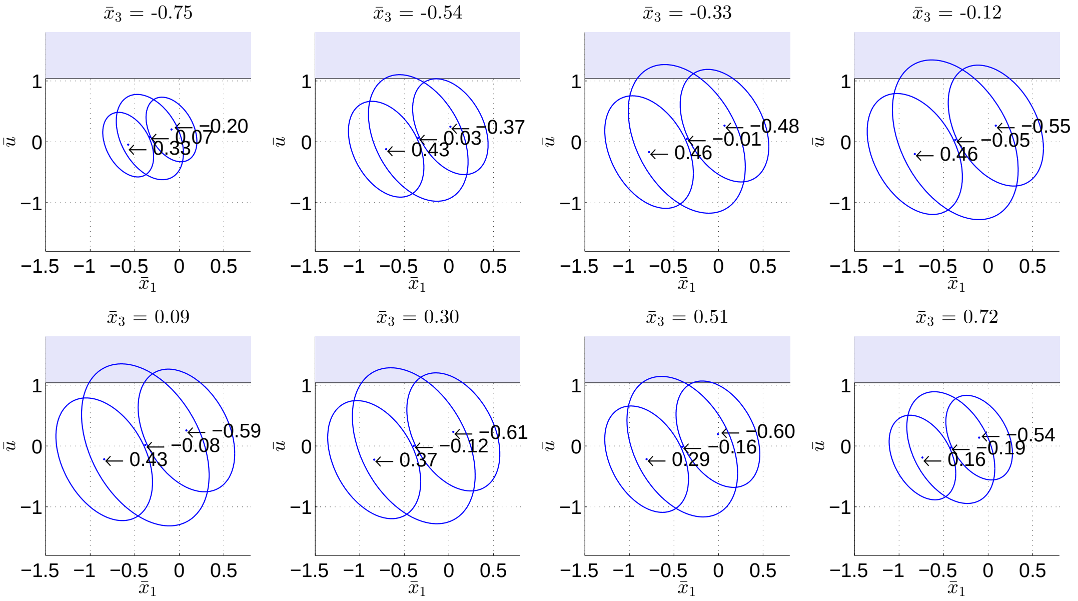

Figure 5: Slices of the synthesized safety-preserving control set as a function of the state when the discriminating kernel is approximated along one terminal direction ℓ. The three ellipsoids in each plot correspond to slices of the set in the augmented space at three values of $\bar{x}_2$ (marked by arrows pointing at the ellipsoids) for fixed $\bar{x}_3$. A scaling value of $\alpha = 0.01$ is used for the inscribed ellipsoids (a value which was experimentally found to maximize the size of the control sets without reducing the kernel approximation significantly).

±80% of the semi-axis that stretches furthest from the center in the negative direction. Also shown (in grey) are the input values outside of $\mathcal{U}$ which will be removed by the intersection operation in (27). Choosing a slice in a subplot fixes $\bar{x}_3$ and $\bar{x}_2$, at which point it is possible to read off a valid range of $\bar{u} \in \mathcal{U}_{\text{ctrl}}(\bar{x})$ for each value of $\bar{x}_1$ (all $\bar{u}$ inside the ellipsoidal slice but outside the grey patch). We note that the set valued control policy represented in this figure may only be applicable at the first time sample because we are working with a finite horizon discriminating kernel. At time sample t_k trajectories may have left $\mathsf{Disc}_{\bar{N}}(\mathcal{K}_0)$ and hence will have to choose from the control policy for $\mathsf{Disc}_{\bar{N}-j}(\mathcal{K}_0)$, where $j \leq k$ by construction. We use (18) to choose j such that $\bar{N} - j = n(x(t_k))$ at each sample time.

To generate the approximation in figure 5 we used a tradeoff factor of $\alpha = 0.01$ in the inscribed ellipsoid optimization problem. We found in our experiments that this value yielded the best results in the sense that the sizes of the discriminating kernel approximation sets as well as the safe control sets appeared largest. For comparison, figure 6 shows the same sets approximated using $\alpha = 0$ (which seeks to maximize the size of the state space kernel with no consideration for the size of the resulting valid input set). As might be expected the safety control sets are significantly smaller in this case; however, the size of the kernel approximation sets (not shown) does not noticeably improve.

4. CONCLUSIONS

The robust sampled data discriminating kernel algorithm described in this paper is an improvement on that from [17] because it is (i) robust to sample time jitter, (ii) tight in the jitter-free case, and (iii) emprically more accurate. We proved the first two properties and demonstrated them on two toy examples, and then used a quadrotor height maintenance problem to provide evidence of the third property and show that the algorithm is practical for real systems.

In the future we plan to further explore how best to handle the tensor product and the intersection operators in the ellipsoidal algorithm, how to compactly represent and efficiently evaluate the sets $\mathcal{K}_k$ and corresponding control policies $\mathcal{U}_{\text{ctrl}}$ for online embedded implementation, and the application of these techniques to more complex, higher dimensional, and hybrid systems. We will also improve the timing model, which presently assumes zero time between data sampling and control signal implementation. A more realistic model should allow a (jittery) time delay between controller input and output.

5. ACKNOWLEDGMENTS

This research was supported by the National Science and Engineering Council of Canada (NSERC) Discovery Grant #298211 (IMM & SK) and NSERC Collaborative Health Research Project #CPG-127770 (SK).

The funding agencies had no role in study design; in the collection, analysis and interpretation of data; in the writing of the report; or in the decision to submit the paper for publication.

6. REFERENCES

[1] A. K. Akametalu, J. F. Fisac, J. H. Gillula, S. Kaynama, M. N. Z. Zeilinger, and C. J. Tomlin. Reachability-based safe online learning with Gaussian processes. In *Proceedings of the IEEE Conference on Decision and Control*, Los Angeles, CA, 2014.

[2] N. Aréchiga and B. Krogh. Using verified control envelopes for safe controller design. In *Proceedings of the American Control Conference*, pages 2918–2923, June 2014.

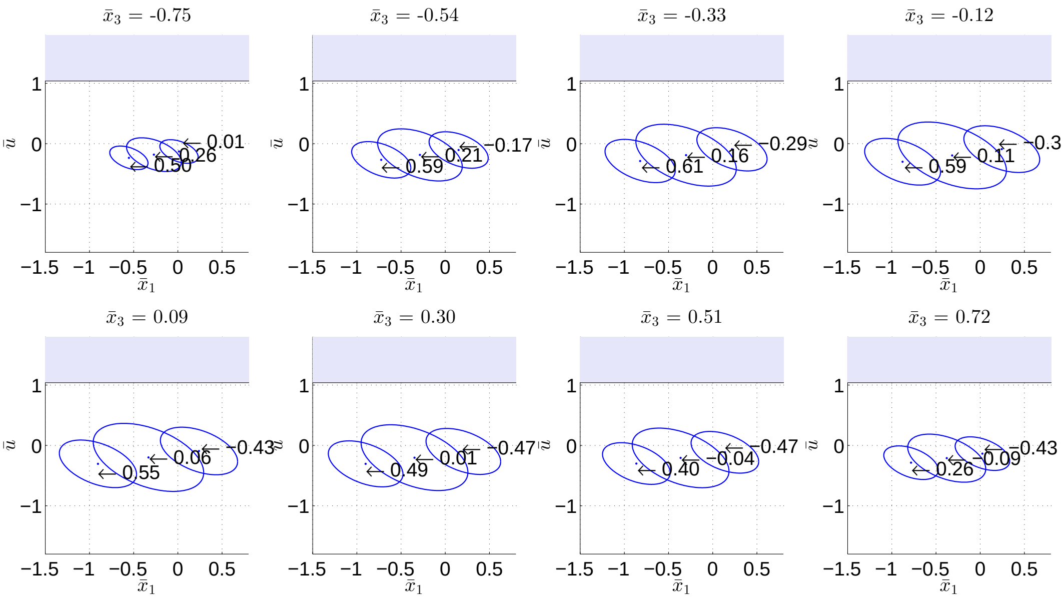

Figure 6: The same synthesized control sets as in Figure 5 but for $\alpha = 0$.

[3] P. Cardaliaguet, M. Quincampoix, and P. Saint-Pierre. Set-valued numerical analysis for optimal control and differential games. In M. Bardi, T. E. S. Raghavan, and T. Parthasarathy, editors, *Stochastic and Differential Games: Theory and Numerical Methods*, volume 4 of *Annals of International Society of Dynamic Games*, pages 177–247. Birkhäuser, 1999.

[4] C. Dabadie, S. Kaynama, and C. J. Tomlin. A practical reachability-based collision avoidance algorithm for sampled-data systems: Application to ground robots. In *International Conference on Intelligent Robots and Systems (IROS)*, 2014.

[5] J. H. Gillula, S. Kaynama, and C. J. Tomlin. Sampling-based approximation of the viability kernel for high-dimensional linear sampled-data systems. In *Hybrid Systems: Computation and Control (HSCC)*, pages 173–182, 2014.

[6] G. C. Goodwin, J. C. Agüero, M. E. Cea Garridos, M. E. Salgado, and J. I. Yuz. Sampling and sampled-data models: The interface between the continuous world and digital algorithms. *IEEE Control Systems Magazine*, 33(5):34–53, Oct 2013.

[7] I. Karafyllis and C. Kravaris. Robust global stabilisability by means of sampled-data control with positive sampling rate. *International Journal of Control*, 82(4):755–772, 2009.

[8] S. Kaynama, J. Maidens, M. Oishi, I. M. Mitchell, and G. A. Dumont. Computing the viability kernel using maximal reachable sets. In *Hybrid Systems: Computation and Control*, pages 55–64, Beijing, China, 2012.

[9] S. Kaynama, I. M. Mitchell, M. M. K. Oishi, and G. A. Dumont. Scalable safety-preserving robust control synthesis for continuous-time linear systems. Submitted February 2013 to IEEE Transactions on Automatic Control, resubmitted December 2013, August 2014.

[10] S. Kaynama, I. M. Mitchell, M. M. K. Oishi, and G. A. Dumont. Safety-preserving control of high-dimensional continuous-time uncertain linear systems. Poster presented at Hybrid Systems Computation and Control (a part of Cyber-Physical Systems Week), April 2013.

[11] A. B. Kurzhanski and P. Varaiya. Ellipsoidal techniques for reachability analysis. In B. Krogh and N. Lynch, editors, *Hybrid Systems: Computation and Control*, number 1790 in Lecture Notes in Computer Science, pages 202–214. Springer Verlag, 2000.

[12] A. B. Kurzhanski and P. Varaiya. Ellipsoidal techniques for reachability analysis: Internal approximation. *Systems and Control Letters*, 41:201–211, 2000.

[13] A. B. Kurzhanski and P. Varaiya. On reachability under uncertainty. *SIAM Journal of Control and Optimization*, 41(1):181–216, 2002.

[14] A. A. Kurzhanskiy and P. Varaiya. Ellipsoidal toolbox. Technical Report UCB/EECS-2006-46, Department of Electrical Engineering and Computer Science, University of California, Berkeley, May 2006.

[15] J. Löfberg. YALMIP : a toolbox for modeling and optimization in MATLAB. In *Computer Aided Control Systems Design*, pages 284–289, September 2004.

[16] I. M. Mitchell. Comparing forward and backward reachability as tools for safety analysis. In A. Bemporad, A. Bicchi, and G. Buttazzo, editors, *Hybrid Systems: Computation and Control*, number 4416 in Lecture Notes in Computer Science, pages 428–443. Springer Verlag, 2007.

[17] I. M. Mitchell, S. Kaynama, M. Chen, and M. Oishi. Safety preserving control synthesis for sampled data systems. *Nonlinear Analysis: Hybrid Systems*, 10:63–82, 2013.

[18] I. M. Mitchell and J. A. Templeton. A toolbox of Hamilton-Jacobi solvers for analysis of nondeterministic continuous and hybrid systems. In M. Morari and L. Thiele, editors, *Hybrid Systems: Computation and Control*, number 3414 in Lecture Notes in Computer Science, pages 480–494. Springer Verlag, 2005.

[19] S. Monaco and D. Normand-Cyrot. Advanced tools for nonlinear sampled-data systems' analysis and control. *European Journal of Control*, 13(2-3):221–241, 2007.

[20] D. Nešić and A. R. Teel. A framework for stabilization of nonlinear sampled-data systems based on their approximate discrete-time models. *IEEE Transactions on Automatic Control*, 49(7):1103–1122, July 2004.

[21] G. Simko and E. K. Jackson. A bounded model checking tool for periodic sample-hold systems. In *Hybrid Systems: Computation and Control (HSCC)*, pages 157–162, 2014.

[22] Y. Tsuchie and T. Ushio. Control-invariance of sampled-data hybrid systems with periodically clocked events and jitter. In *Proceedings of the IFAC Conference on Analysis and Design of Hybrid Systems*, pages 417–422, 2006.